# The RATS Control Protocol (RCP)

*Perry A. Blackmore*

**Communications Division**
**Electronics and Surveillance Research Laboratory**

DSTO-TN-0355

## ABSTRACT

RATS is the real-time scheduler used in the server of the DSTO Theatre Broadcast System demonstrator. This document describes the RATS Control Protocol which is used for all communications with RATS. RCP is based on the User Datagram Protocol (UDP) and is used for all user requests, management requests, and control activity performed by RATS.

**RELEASE LIMITATION**

*Approved for public release*

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION DSTO

*Published by*

*DSTO Electronics and Surveillance Research Laboratory*
*PO Box 1500*
*Salisbury South Australia 5108 Australia*

*Telephone: (08) 8259 5555*
*Fax: (08) 8259 6567*

*AR-011-854*
*May 2001*

**APPROVED FOR PUBLIC RELEASE**

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION DSTO

# The RATS Control Protocol (RCP)

Perry A. Blackmore

DSTO-TN-0355

# The RATS Control Protocol (RCP)

## Executive Summary

At the heart of the DSTO Theatre Broadcast System (TBS) information management system is an application called RATS (Real Time Scheduler). RATS is responsible for managing the flow of traffic broadcast over the TBS. Its main functions include accepting scheduling requests from users and other system components and performing real time optimisations in order to deliver the maximum military utility of information broadcast. This report describes a protocol developed for communications to and from RATS called the RATS Control Protocol (RCP). RCP is used by users to submit scheduling requests to RATS, by agents to perform management functions on RATS, and by RATS to deliver control information to various entities within the TBS server.

# Contents

# 1. Introduction

DSTO, under Joint Project 2008 Phase 3C, has developed a Theatre Broadcast System (TBS) Demonstrator. It is based on commercial hardware (MPEG encoders, Integrated Receive Devices, Digital Video Broadcast modulators), military grade encrytion devices (KIV-7, KIV-19) and DSTO developed software. One of the key software components is RATS (Real Time Scheduler) which is responsible for scheduling all information to be passed over the TBS.

RATS accepts requests from users and other system components and performs a real time optimisation in order to deliver the maximum military utility of information broadcast. RATS implements the schedule by communicating it to the system applications. All communications to and from RATS are done via a protocol called the RATS Control Protocol (RCP). This document describes this protocol.

## 1.1 Background

Work on RATS began in September 1996 under Project Awareness. It was initially intended as a tool to investigate QoS issues in heterogenous networks. It quickly became evident that RATS was ideal from controlling streams on broadcast networks. A UDP (User Datagram Protocol) based protocol was developed for communications with RATS which would evolve into RCP. The initial intention of the development of RCP was to provide a protocol to be used solely for communications to and from RATS. However it is now used for communication between most entities within the TBS demonstrator system.

RCP has evolved through versions 0.0, 0.1, 1.0, 1.1, and 1.2. The version 1.1 was fixed in October 1998, and the current version 1.2 was fixed in March 2000.

## 1.2 Future Work

A complete rewrite of RCP is desirable at a future date. This is required because:

1. RCP has developed in a rather adhoc manner and as a result a number of inefficiencies and redundancies exist in the current version, and
2. A TCP based protocol has been developed for communications between client applications.

The new protocol would be suitable for operation over TCP (Transport Control Protocol) and UDP. This protocol would be used by all entities within the TBS.

# 2. The RCP Protocol

The RCP protocol (version 1.2) is described in this section. Note that all packet and message coding is according to network byte order (MSB order).

Packet Format

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>9</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td colspan="8">... PASSWORD ...</td><td colspan="8">VERSION</td></tr>
<tr><td colspan="16">CATEGORY</td></tr>
<tr><td colspan="8">... MSG 1 ...</td><td colspan="8">... MSG 2 ...</td></tr>
<tr><td colspan="8">...</td><td colspan="8">... MSG N ...</td></tr>
<tr><td colspan="16">CRC</td></tr>
<tr><td colspan="16"></td></tr>
</table>

PASSWORD: variable

RCP password. Each RCP packet is encoded in one UDP packet. Each RCP packet commences with a password to avoid conflicting with other packets that may be on the network. Currently the password used is "smartfish".

VERSION: 8 bits

Version number. The version is in the form of "x.y" where "x" is encoded in the most significant 4 bits (bit positions 8 – 11 in the above diagram) and "y" is encoded in the least significant 4 bits (12 – 15). For example "00010010" would represent RCP version 1.2.

CATEGORY: 16 bits

Device category. A 2 octet bit-field specifies the type of device that the RCP packet is intended for. The codes are:

| bit position | category code | Category description |
|---|---|---|
| 15 | rcpCAT_SCHEDULER = 0 | scheduler |
| 14 | rcpCAT_PROXY = 1 | proxy |
| 14 | rcpCAT_REQUESTOR = 2 | requestor |
| 13 | rcpCAT_MANAGER = 3 | manager |
| 12 | rcpCAT_FILE_DEVICE = 4 | file transfer device |
| 11 | rcpCAT_AUDIO_DEVICE = 5 | audio device |
| 10 | rcpCAT_VIDEO_DEVICE= 6 | video device |
| 9 | rcpCAT_REPLAY_DEVICE = 7 | videoclip replay device |
| 8 | rcpCAT_STREAM_DEVICE = 8 | stream based device |
| 7...0 | reserved | |

MSG: variable

RCP messages. Each RCP packet contains one or more RCP variable length messages. These follow the CATEGORY field.

ERROR: 8 bits
Error condition. Encoded as an unsigned 8 bit integer.

A type 1 RESPONSE message has the following structure:

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>9</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td colspan="8">OPCODE</td><td colspan="8">HANDLE</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td colspan="8"></td><td colspan="8">TYPE</td></tr>
<tr><td colspan="8">RESPONSE</td><td colspan="8">ERROR</td></tr>
<tr><td colspan="16">ARG 1</td></tr>
<tr><td colspan="16"></td></tr>
</table>

OPCODE, HANDLE, TYPE, RESPONSE, ERROR
As per RESPONSE type 0 message.

ARG 1: 32 bits
Unsigned 32 bit integer.

A type 2 RESPONSE message has the following structure:

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>9</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td colspan="8">OPCODE</td><td colspan="8">HANDLE</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td colspan="8"></td><td colspan="8">TYPE</td></tr>
<tr><td colspan="8">RESPONSE</td><td colspan="8">ERROR</td></tr>
<tr><td colspan="8">ARG 1</td><td colspan="8">NULL</td></tr>
</table>

OPCODE, HANDLE, TYPE, RESPONSE, ERROR
As per RESPONSE type 0 message.

ARG 1: variable
Character array.

NULL: 8 bits
Zero field used to terminate ARG 1.

A type 3 RESPONSE message has the following structure:

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>9</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td colspan="8">OPCODE</td><td colspan="8">HANDLE</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td colspan="8"></td><td colspan="8">TYPE</td></tr>
<tr><td colspan="8">RESPONSE</td><td colspan="8">ERROR</td></tr>
<tr><td colspan="16">ARG 1</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td colspan="16">ARG 2</td></tr>
</table>

OPCODE, HANDLE, TYPE, RESPONSE, ERROR
As per RESPONSE type 0 message.

ARG 1: 32 bits
Unsigned 32 bit integer.

ARG 2: 16 bits
Unsigned 16 bit integer.

A type 4 RESPONSE message has the following structure:

| 0 1 2 3 4 5 6 7 | 9 9 10 11 12 13 14 15 |
|---|---|
| OPCODE | HANDLE |
| | |
| | TYPE |
| RESPONSE | ERROR |
| ARG 1 | |
| | |
| ARG 2 | |
| | |
| ARG 3 | |
| | |
| ARG 4 | |

OPCODE, HANDLE, TYPE, RESPONSE, ERROR
As per RESPONSE type 0 message.

ARG 1: 32 bits
Unsigned 32 bit integer.

ARG 2: 32 bits
Real.[1]

ARG 3: 32 bits
Unsigned 32 bit integer.

ARG 4: 16 bits
Unsigned 16 bit integer.

A type 5 RESPONSE message has the following structure:

[1] To enable cross platform compatibility real numbers are encoded as signed 32 bit integers by first multiplying by 1000 and truncating.

| 0 1 2 3 4 5 6 7 | 9 9 10 11 12 13 14 15 |
|---|---|
| OPCODE | HANDLE |
| | |
| | TYPE |
| RESPONSE | ERROR |
| LENGTH | |
| ... ARG 1 ... | |
| ARG 2 | |
| | |
| ARG 3 | |

OPCODE, HANDLE, TYPE, RESPONSE, ERROR
As per RESPONSE type 0 message.

LENGTH: 16 bits
Length of ARG 1 field. Encoded as a 16 bit unsigned integer.

ARG 1: variable
Character array.

ARG 2: 32 bits
Unsigned 32 bit integer.

ARG 3: 16 bits
Unsigned 16 bit integer.

### 2.1.3 INIT Message

The INIT message is used by RATS to initialise a data stream within an application, for example to request a file transfer application to prepare to deliver a file.

| 0 1 2 3 4 5 6 7 | 9 9 10 11 12 13 14 15 |
|---|---|
| OPCODE | HANDLE |
| | |
| | TXS |
| ... LOC_NAME ... | NULL |
| ... REM_NAME ... | NULL |

OPCODE, HANDLE
As per NULL message.

TXS: 8 bits
The number of transmissions required. Encoded as an unsigned 8 bit integer.

LOC_NAME: variable

Character array specifying local name of file to be delivered.

NULL: 8 bits
Zero field used to terminate character arrays.

REM_NAME: variable
Character array specifying remote name of file to be delivered.

### 2.1.4 CLOSE Message

The INIT message is used by RATS to initialise a data stream within an application, for example to request a file transfer application to prepare to deliver a file.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 9 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OPCODE | | | | | | | | HANDLE | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

OPCODE, HANDLE
As per NULL message.

### 2.1.5 RATE Message

The RATE message is used by RATS to set the data rate of a data stream within an application.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 9 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OPCODE | | | | | | | | HANDLE | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | RATE | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

OPCODE, HANDLE
As per NULL message.

RATE: 32 bits
New rate for data stream. Encoded as an unsigned 32 bit integer.

### 2.1.6 FILE_DELIVER Message

The FILE_DELIVER message is used by RATS to request the delivery of a file by a file transfer application. It has not been used in versions of RATS beyond 2.0.

| 0 1 2 3 4 5 6 7 | 9 9 10 11 12 13 14 15 |
|---|---|
| OPCODE | HANDLE |
|  |  |
|  | IP |
|  |  |
|  | PORT |
|  | ... LOC_NAME ... |
| NULL | ... REM_NAME ... |
| NULL |  |

OPCODE, HANDLE
As per NULL message.

IP: 32 bits
The IP address of the destination host. Encoded as an unsigned 32 bit integer.

PORT: 16 bits
The UDP port of the destination host. Encoded as an unsigned 16 bit integer.

LOC_NAME: variable
Character array specifying local name of file to be delivered.

NULL: 8 bits
Zero field used to terminate character arrays.

REM_NAME: variable
Character array specifying remote name of file to be delivered.

### 2.1.7 FRAME_RATE Message

The FRAME_RATE message is used by RATS to set the frame rate of a video stream

| 0 1 2 3 4 5 6 7 | 9 9 10 11 12 13 14 15 |
|---|---|
| OPCODE | HANDLE |
|  |  |
|  | RATE |
|  |  |

OPCODE, HANDLE
As per NULL message.

RATE: 16 bits
New frame rate for video stream. Encoded as an unsigned 16 bit integer.

### 2.1.8 ENCODER Message

The ENCODER message is used by RATS to set the coding scheme for a real-time application.

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>9</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="8">OPCODE</td><td colspan="8">HANDLE</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td colspan="8"></td><td colspan="8">CODE</td></tr>
</table>

OPCODE, HANDLE
As per NULL message.

CODE: 8 bits
New coding scheme. The encoder types and their values are as follows:

| value | encoder code | code description |
|---|---|---|
| 0 | rcpCOD_H261 | h261 video encoding |
| 1 | rcpCOD_JPEG | JPEG video encoding |
| 2 | rcpCOD_NV | NV video encoding |
| 3 | rcpCOD_NVDCT | NVDCT video encoding |
| 4 | rcpCOD_CELLB | CELLB video encoding |
| 5 | rcpCOD_PCM | PCM audio encoding |
| 6 | rcpCOD_PCM2 | PCM audio encoding (version 2) |
| 7 | rcpCOD_PCM4 | PCM audio encoding (version 4) |
| 8 | rcpCOD_DVI | DVI audio encoding |
| 9 | rcpCOD_DVI2 | DVI audio encoding (version 2) |
| 10 | rcpCOD_DVI4 | DVI audio encoding (version 4) |
| 11 | rcpCOD_GSM | GSM audio encoding |
| 12 | rcpCOD_LPC4 | LPC audio encoding |

### 2.1.9 QUALITY Message

The QUALITY message is used by RATS to set the quality of a video stream

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>9</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="8">OPCODE</td><td colspan="8">HANDLE</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td colspan="8"></td><td colspan="8">QUAL</td></tr>
<tr><td colspan="8"></td></tr>
</table>

OPCODE, HANDLE
As per NULL message.

QUAL: 16 bits
New quality for video stream. Encoded as an unsigned 16 bit integer.

### 2.1.10 REQUEST Message

The REQUEST message is used to submit scheduling requests to RATS.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 9 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OPCODE | | | | | | | | HANDLE | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | USER | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | ID | | | | TYPE | | | |
| VALUE | | | | | | | | IP | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | PORT | | | | | | | |
| | | | | | | | | START | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | .... X ... | | | | | | | |

OPCODE, HANDLE

As per NULL message.

USER: 32 bits

Hashed user name and password. The user name and password are appended and then a CRC-32 is calculated. Encoded as an unsigned 32 bit integer.

ID: 4 bits

Application identifier.

TYPE: 4 bits

Application type. This is encoded as follows:

| Application type | Code |
|---|---|
| NULL | 0 - 3 |
| FILE_DEVICE | rcpCAT_FILE_DEVICE = 4 |
| AUDIO_DEVICE | rcpCAT_AUDIO_DEVICE = 5 |
| VIDEO_DEVICE | rcpCAT_VIDEO_DEVICE = 6 |
| REPLAY_DEVICE | rcpCAT_REPLAY_DEVICE = 7 |

VALUE: 8 bits

User perceived value of request. Encoded as an unsigned 8 bit integer.

IP: 32 bits

The IP address of the requesting agent. Encoded as an unsigned 32 bit integer.

PORT: 16 bits

The UDP port of the requesting agent. Encoded as an unsigned 16 bit integer.

START: 32 bits

The requested start time. Encoded as a real.

X: variable

Fields dependent on TYPE.

For TYPE = FILE_DEVICE, X has the form:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 9 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ... LOC_NAME ... | | | | | | | | NULL | | | | | | | |
| ... REM_NAME ... | | | | | | | | NULL | | | | | | | |
| TML | | | | PREC | | | | SIZE | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

LOC_NAME: variable

Character array specifying local name of file to be delivered.

NULL: 8 bits

Zero field used to terminate character arrays.

REM_NAME: variable

Character array specifying remote name of file to be delivered.

TML: 4 bits

Timeliness requirement of file request. This is encoded as follows:

| Timeliness | Code |
|---|---|
| NO_SLACK | 0 |
| SOME_SLACK | 1 |
| LOTS_OF_SLACK | 2 |

PREC: 4 bits

Precedence of file request. This is encoded as follows:

| Precedence | Code |
|---|---|
| ROUTINE | 0 |
| PRIORITY | 1 |
| IMMEDIATE | 2 |
| FLASH | 3 |

SIZE: 32 bits

Size of requested file in bytes. Encoded as an unsigned 32 bit integer.

For TYPE = AUDIO_DEVICE and TYPE = VIDEO_DEVICE, X has the form:

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>9</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="16">LENGTH</td></tr>
<tr><td colspan="16"></td></tr>
</table>

LENGTH: 32 bits

Length is the requested time for broadcast of the real-time stream. Encoded as an unsigned 32 bit integer.

For TYPE = REPLAY_DEVICE, X has the form:

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>9</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="8">... NAME ...</td><td colspan="8">NULL</td></tr>
<tr><td colspan="16">RATE</td></tr>
<tr><td colspan="16"></td></tr>
</table>

NAME: variable

Character array specifying local name of file to be replayed.

NULL: 8 bits

Zero field used to terminate character arrays.

RATE: 32 bits

Rate at which requested file is to be replayed at. Encoded as an unsigned 32 bit integer.

### 2.1.11 KILL Message

The KILL message is used to request a task be removed from the RATS schedule.

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>9</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="8">OPCODE</td><td colspan="8">HANDLE</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td colspan="8"></td><td colspan="8">USER</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td colspan="8"></td><td colspan="8">IP</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td colspan="8"></td><td colspan="8">PORT</td></tr>
<tr><td colspan="8"></td></tr>
</table>

OPCODE, HANDLE

As per NULL message.

USER: 32 bits

Hashed user name and password. The user name and password are appended and then a CRC-32 is calculated. Encoded as an unsigned 32 bit integer.

IP: 32 bits

The IP address of the requesting agent. Encoded as an unsigned 32 bit integer.

PORT: 16 bits

The UDP port of the requesting agent. Encoded as an unsigned 16 bit integer.

### 2.1.12 LOGIN Message

The LOGIN message is used to remotely login to RATS to perform management operations.

| 0 1 2 3 4 5 6 7 | 9 9 10 11 12 13 14 15 |
|---|---|
| OPCODE | HANDLE |
| | |
| | USER |
| | |
| | IP |
| | |
| | PORT |
| | |

OPCODE, HANDLE, USER, IP, PORT

As per KILL message.

### 2.1.13 LOGOUT Message

The LOGOUT message is used to remotely logout of RATS after a logging in.

| 0 1 2 3 4 5 6 7 | 9 9 10 11 12 13 14 15 |
|---|---|
| OPCODE | HANDLE |
| | |
| | USER |
| | |
| | IP |
| | |
| | PORT |
| | |

OPCODE, HANDLE, USER, IP, PORT

As per KILL message.

### 2.1.14 GET_ACCOUNTS Message

The GET_ACCOUNTS message is used to request RATS to send the current user accounts to the requesting agent.

| 0 1 2 3 4 5 6 7 | 9 9 10 11 12 13 14 15 |
|---|---|
| OPCODE | HANDLE |
| | |
| | USER |
| | |
| | IP |
| | |
| | PORT |
| | |

OPCODE, HANDLE, USER, IP, PORT
As per KILL message.

### 2.1.15 SET_ACCOUNTS Message

The SET_ACCOUNTS message is used to request RATS to update the current user accounts.

| 0 1 2 3 4 5 6 7 | 9 9 10 11 12 13 14 15 |
|---|---|
| OPCODE | HANDLE |
| | |
| | USER |
| | |
| | IP |
| | |
| | PORT |
| | LENGTH |
| | ... ACCOUNTS ... |

OPCODE, HANDLE, USER, IP, PORT
As per KILL message.

LENGTH: 16 bits
Length of ACCOUNTS field. Encoded as a 16 bit unsigned integer.

ACCOUNTS: variable
Character array with accounts information.

### 2.1.16 COMPLETE Message

The COMPLETE message is used to inform applications of the completion of a task. It has been used by MUSTAFA to inform WEB_AGENT of the arrival of a file.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 9 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OPCODE | | | | | | | | HANDLE | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | STATUS | | | | | | | |
| ... NAME ... | | | | | | | | NULL | | | | | | | |

OPCODE, HANDLE
As per NULL message.

STATUS: 8 bits
Field to specify the status of the task at completion. Encoded as an 8 bit unsigned integer. Currently 0 = unsuccessful completion, 1 = successful completion.

NAME: variable
Character array with task information such as a file name.

NULL: 8 bits
Zero field used to terminate NAME.

### 2.1.17 GET_PARAMS Message

The GET_PARAMS message is used by RATS to request MUSTAFA for its current operational parameters.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 9 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OPCODE | | | | | | | | HANDLE | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

OPCODE, HANDLE
As per NULL message.

### 2.1.18 GET_PROG Message

The GET_PROG message is used by RATS to request MUSTAFA for its current file status.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 9 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OPCODE | | | | | | | | HANDLE | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | TIME | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

OPCODE, HANDLE

As per NULL message.

TIME: 64 bits

Time to send file progress express in elapsed seconds since 00:00 Universal Coordinated Time, January 1, 1970. Encoded as a double precision real. Currently not used.

### 2.1.19 PARAMS Message

The PARAMS message is used by MUSTAFA to respond to a GET_PARAMS request.

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>9</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td colspan="8">OPCODE</td><td colspan="8">HANDLE</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td colspan="8"></td><td colspan="8">PL</td></tr>
<tr><td colspan="8"></td><td colspan="8">FO</td></tr>
<tr><td colspan="8"></td><td colspan="8">DA</td></tr>
<tr><td colspan="8"></td><td colspan="8">PO</td></tr>
<tr><td colspan="8"></td></tr>
</table>

OPCODE, HANDLE

As per NULL message.

PL: 16 bits

Payload length used in MUSTAFA data packets. Encoded as a unsigned 16 bit integer.

FO: 16 bits

File information overhead in MUSTAFA file information packets. Encoded as a unsigned 16 bit integer.

DA: 16 bits

Data overhead in MUSTAFA data packets. Encoded as a unsigned 16 bit integer.

PO: 16 bits

Layer 1 & 2 overhead in MUSTAFA packets. Encoded as a unsigned 16 bit integer.

### 2.1.20 PROG Message

The PROG message is used by MUSTAFA to respond to a GET_PROG request.

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>9</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="8">OPCODE</td><td colspan="8">HANDLE</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td colspan="8"></td><td colspan="8">NFILES</td></tr>
<tr><td colspan="8"></td><td colspan="8">ID 1</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td colspan="8"></td><td>F 1</td><td colspan="7">BYTES 1</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td colspan="8"></td><td colspan="8">ID 2</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td colspan="8"></td><td>F 2</td><td colspan="7">BYTES 2</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td colspan="8"></td><td colspan="8">...</td></tr>
<tr><td colspan="16">...</td></tr>
<tr><td colspan="8"></td><td colspan="8">ID N</td></tr>
<tr><td colspan="8"></td><td colspan="8"></td></tr>
<tr><td colspan="8"></td><td>F N</td><td colspan="7">BYTES N</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td colspan="8"></td></tr>
</table>

OPCODE, HANDLE
As per NULL message.

NFILES: 16 bits
The number of files which progress information is supplied. Encoded as an unsigned 16 bit integer.

NFILES: 16 bits
Payload length used in MUSTAFA data packets. Encoded as an unsigned 16 bit integer.

ID: 32 bits
File identifier. Encoded as an unsigned 32 bit integer.

F: 1 bit
File completion indicator. If bit is set then the file has completed.

BYTES: 31 bit
Number of bytes left to transmit for given file. Encoded as an unsigned 31 bit integer.

### 2.1.21 TRACE Message

The TRACE message is used to set the output trace level of various TBS applications including RATS.

| 0 1 2 3 4 5 6 7 | 9 9 10 11 12 13 14 15 |
|---|---|
| OPCODE | HANDLE |
| | |
| | … TR … |
| NULL | |

OPCODE, HANDLE
As per NULL message.

TR: variable
Character array with trace string.

NULL: 8 bits
Zero field used to terminate TR.

### 2.1.22 STREAM Message

The STREAM message is used by RATS to establish a new stream in a MUSTAFA process.

| 0 1 2 3 4 5 6 7 | 9 9 10 11 12 13 14 15 |
|---|---|
| OPCODE | HANDLE |
| | |
| | LOC_PORT |
| | REM_PORT |
| | REM_HOST |
| | |
| | |

OPCODE, HANDLE
As per NULL message.

LOC_PORT: 16 bits
The UDP port for the stream on the local host. Encoded as an unsigned 16 bit integer.

REM_PORT: 16 bits
The UDP port for the stream on the destination host. Encoded as an unsigned 16 bit integer.

REM_HOST: 32 bits
The IP address of the stream destination host. Encoded as an unsigned 32 bit integer.

### 2.1.23 TERMINATE Message

The TERMINATE message is used to terminate a process.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 9 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OPCODE | | | | | | | | HANDLE | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

OPCODE, HANDLE

As per NULL message.

DISTRIBUTION LIST

The RATS Control Protocol (RCP)

Perry A Blackmore

**AUSTRALIA**

**DEFENCE ORGANISATION**

**Task Sponsor** Director General C3I Development

**S&T Program**

Chief Defence Scientist
FAS Science Policy shared copy
AS Science Corporate Management
Director General Science Policy Development
Counsellor Defence Science, London (Doc Data Sheet)
Counsellor Defence Science, Washington (Doc Data Sheet)
Scientific Adviser to MRDC Thailand (Doc Data Sheet)
Scientific Adviser Policy and Command
Navy Scientific Adviser
Scientific Adviser - Army
Air Force Scientific Adviser
Director Trials

**Aeronautical and Maritime Research Laboratory**
Director

**Electronics and Surveillance Research Laboratory**
Director (Doc Data Sheet and distribution list only)

Chief of Communications Division
Research Leader, Military Information Networks
Head Network Architectures Group
Perry Blackmore (3 copies)
Task Manager - Philip Stimson

**DSTO Library**
Library Fishermans Bend
Library Maribyrnong (Doc Data Sheet only)
Library Salisbury
Australian Archives
Library, MOD, Pyrmont (Doc Data sheet only)
US Defense Technical Information Center, 2 copies
UK Defence Research Information Centre, 2 copies
Canada Defence Scientific Information Service, 1 copy
NZ Defence Information Centre, 1 copy
National Library of Australia, 1 copy

**Capability Systems Staff**

Director General Maritime Development (Doc Data Sheet only)
Director General Aerospace Development (Doc Data Sheet only)

**Knowledge Staff**

Director General Command, Control, Communications and Computers (DGC4) (Doc Data Sheet only)
Director General Intelligence, Surveillance, Reconnaissance, and Electronic Warfare (DGISREW)R1-3-A142 CANBERRA ACT 2600 (Doc Data Sheet only)
Director General Defence Knowledge Improvement Team (DGDKNIT) R1-5-A165, CANBERRA ACT 2600 (Doc Data Sheet only)

**Army**

ASNSO ABCA, Puckapunyal (4 copies)
SO (Science), DJFHQ(L), MILPO Enoggera, Queensland 4051 (Doc Data Sheet only)
NAPOC QWG Engineer NBCD c/- DENGRS-A, HQ Engineer Centre Liverpool Military Area, NSW 2174 (Doc Data Sheet only)

**Intelligence Program**

DGSTA Defence Intelligence Organisation
Manager, Information Centre, Defence Intelligence Organisation

**Corporate Support Program**

OIC TRS, Defence Regional Library, Canberra
Officer in Charge, Document Exchange Centre (DEC), 1 copy

**UNIVERSITIES AND COLLEGES**

Australian Defence Force Academy
Library
Head of Aerospace and Mechanical Engineering
Senior Librarian, Hargrave Library, Monash University
Librarian, Flinders University

**OTHER ORGANISATIONS**

NASA (Canberra)
Info Australia (formerly AGPS)
State Library of South Australia
Parliamentary Library, South Australia

## OUTSIDE AUSTRALIA

**ABSTRACTING AND INFORMATION ORGANISATIONS**

INSPEC: Acquisitions Section Institution of Electrical Engineers
Library, Chemical Abstracts Reference Service
Engineering Societies Library, US
Materials Information, Cambridge Scientific Abstracts, US
Documents Librarian, The Center for Research Libraries, US

**INFORMATION EXCHANGE AGREEMENT PARTNERS**

Acquisitions Unit, Science Reference and Information Service, UK
Library - Exchange Desk, National Institute of Standards and Technology, US

SPARES (5 copies)

**Total number of copies: 53**

## DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA

1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)

2. TITLE

The RATS Control Protocol (RCP)

3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)

| | |
|---|---|
| Document | (U) |
| Title | (U) |
| Abstract | (U) |

4. AUTHOR(S)

Perry A. Blackmore

5. CORPORATE AUTHOR

Electronics and Surveillance Research Laboratory
PO Box 1500
Salisbury SA 5108 Australia

| | | | |
|---|---|---|---|
| 6a. DSTO NUMBER<br>DSTO-TN-0355 | 6b. AR NUMBER<br>AR-011-854 | 6c. TYPE OF REPORT<br>Technical Note | 7. DOCUMENT DATE<br>May 2001 |

| | | | | |
|---|---|---|---|---|
| 8. FILE NUMBER<br>E8709/4/15/2 | 9. TASK NUMBER<br>99/140 | 10. TASK SPONSOR<br>DGFD (Joint) | 11. NO. OF PAGES<br>20 | 12. NO. OF REFERENCES<br>0 |

13. URL ON WORLDWIDE WEB

http://www.dsto.defence.gov.au/corporate/reports/DSTO-TN-0355.pdf

14. RELEASE AUTHORITY

Chief, Communications Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

*Approved for public release*

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE CENTRE, DIS NETWORK OFFICE, DEPT OF DEFENCE, CAMPBELL PARK OFFICES, CANBERRA ACT 2600

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CASUAL ANNOUNCEMENT Yes

18. DEFTEST DESCRIPTORS

Real time data processing, Decision making - data processing, Communications networks, Scheduling, Military communications

19. ABSTRACT

RATS is the real-time scheduler used in the server of the DSTO Theatre Broadcast System demonstrator. This document describes the RATS Control Protocol which is used for all communications with RATS. RCP is based on the User Datagram Protocol (UDP) and is used for all user requests, management requests, and control activity performed by RATS.

TECHNICAL NOTE DSTO-TN-0355 AR-011-854 MAY 2001

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION
DSTO

ELECTRONICS AND SURVEILLANCE RESEARCH LABORATORY
PO BOX 1500 SALISBURY SOUTH AUSTRALIA, 5108
AUSTRALIA, TELEPHONE (08) 8259 5555